Российская Федерация
Министерство образования Калининградской области

Государственное автономное учреждение Калининградской области дополнительного профессионального образования

«Институт развития образования»

236016, г. Калининград, ул. Томская, 19
тел/факс: (4012) 578-301
e-mail: info@koiro.edu.ru
www.koiro.edu.ru

ОГРН 1023901014323
ИНН 3906020548

Дополнительная общеобразовательная
общеразвивающая программа
«Углубленное изучение информатики. Подготовка к сдаче единого государственного экзамена (11 класс)»

Возраст обучающихся 16-18 лет
Срок реализации – 2 года
(288 часов)

Программа обсуждена и утверждена
на заседании Ученого совета
от 31.01.2018 г. (Протокол № 1)

Председатель Ученого совета

/Л. А. Зорькина/

Калининград
2018

ЛИСТ СОГЛАСОВАНИЯ

Составитель: Скабицкая Юлия Александровна, методист учебно-методического центра управления образованием Калининградского областного института развития образования.

Дополнительная общеобразовательная общеразвивающая программа «Углубленное изучение информатики. Подготовка к сдаче единого государственного экзамена (11 класс)» обсуждена и утверждена на заседании Центра информатизации образования Калининградского областного института развития образования (протокол № 1 от 29 января 2018 года).

Начальник Центра информатизации образования _____ /Д. Ю.Кулагин/

Дополнительная общеобразовательная общеразвивающая программа «Углубленное изучение информатики. Подготовка к сдаче единого государственного экзамена (11 класс)» одобрена Ученым советом Калининградского областного института развития образования (Протокол № 1 от «31» января 2018 г.).

Программа пересмотрена на заседании Ученого совета_____
_____
_____

Внесены следующие изменения (или изменений не внесено):
_____
_____
_____
_____

Протокол №____от «____»_____ 201__ г.

Проректор по научно-методической работе

_____ /В. П. Вейдт/

СОДЕРЖАНИЕ

дополнительной общеобразовательной общеразвивающей программы **«Углубленное изучение информатики. Подготовка к сдаче единого государственного экзамена (11 класс)»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Актуальность данной программы объясняется возросшим значением предмета «Информатика» в формировании междисциплинарных связей, причем как на уровне понятийного аппарата, так и на уровне инструментария. Многие положения, развиваемые информатикой, рассматриваются как основа создания и использования информационных и коммуникационных технологий – одного из наиболее значимых технологических достижений современной цивилизации. Вместе с математикой, физикой, химией, биологией курс информатики закладывает основы технического и естественнонаучного мировоззрения.

Одной из основных черт нашего времени является всевозрастающая изменчивость окружающего мира. В этих условиях велика роль освоения предметных областей, обеспечивающих профессиональную мобильность человека, готовность его к освоению новых технологий, в том числе информационных. Необходимость подготовки личности к быстро наступающим переменам в обществе требует развития разнообразных форм мышления, формирования у учащихся умений организации собственной учебной деятельности, их ориентации на деятельностную жизненную позицию.

В содержании программы «Углубленное изучение информатики. Подготовка к сдаче единого государственного экзамена (11 класс)» основной школы акцент сделан на изучении математических основ информатики, формировании информационной культуры, развитии алгоритмического мышления, освоении навыков программирования на одном из современных высокоуровневых языков.

Персонализация дополнительного образования усиливает его преимущества по сравнению с другими институтами формального образования посредством актуализации таких аспектов как возможность выбора режима и темпа освоения образовательных программ, выстраивания индивидуальных образовательных траекторий.

Обучение в рамках данной программы возможно с разновозрастным составом обучающихся от 16 до 18 лет.

Особенностью программы «Углубленное изучение информатики. Подготовка к сдаче единого государственного экзамена (11 класс)» является многоуровневость, реализованная с помощью блочно-модульной системы, которая позволяет учитывать различные уровни базовой подготовки обучающихся. Программа составлена таким образом, чтобы обучающиеся имели возможность выборочно освоить учебный материал – в случае обнаружения дефицитов по отдельным темам учебного курса предмета «Информатика».

Программа реализует педагогическую идею формирования у обучающихся умения учиться самостоятельно, добывать и систематизировать новые знания.

Образовательный процесс, а также реализация индивидуального образовательного маршрута реализуется с использованием дистанционных образовательных технологий. Для организации автоматизированной проверки решений обучающихся используются ресурсы дистанционного сервиса http://informatics.mccme.ru/.

Программа обеспечивает реализацию следующих **принципов**:

– непрерывность дополнительного образования;

– развития индивидуальности каждого ребенка в процессе социального самоопределения;

– системность организации учебно-воспитательного процесса;

– раскрытие способностей и поддержка одаренности детей.

Направленность программы – техническая.

Реализация дополнительной профессиональной программы «Углубленное изучение информатики. Подготовка к сдаче единого государственного экзамена (11 класс)» направлено на достижение следующих **целей**:

1) овладение умениями применять, анализировать, преобразовывать информационные модели реальных объектов и процессов, используя при этом информационные и коммуникационные технологии (ИКТ);

2) развитие познавательных интересов, интеллектуальных и творческих способностей путем освоения и использования методов информатики и средств ИКТ при изучении различных учебных предметов;

3) воспитание ответственного отношения к соблюдению этических и правовых норм информационной деятельности;

4) приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.

Значительное место в программе отводится знакомству обучающихся со способами оптимизации математических и алгоритмических методов решения заданий, входящих в состав единого государственного экзамена по предмету «Информатика», а также изучению классических алгоритмов обработки информации.

Педагогическая целесообразность программы состоит в применении технологии дифференцированного обучения, которая является наиболее эффективной формой индивидуализации учебного процесса, обеспечивающая максимально благоприятные условия для обучающихся.

Академический объем программы составляет 96 часов в первый год обучения и 192 часа во второй год обучения. Продолжительность занятий – 45 минут.

Структура образовательной программы построена по модульному принципу и представлена инвариантными модулями, каждый из которых знакомит обучающихся с методами решения соответствующих теме модуля заданий, водящих в экзаменационную работу.

Тема инвариантного модуля первого уровня обучения обязательно изучается всеми учащимися в течение первого года. На этом этапе обучения учащиеся получают общее представление о синтаксисе и базовых операторах языка программирования Python, а также об основных алгоритмах обработки данных. Второй этап обучения позволяет обучающимся систематизировать знания, полученные при изучении предмета «Информатика», освоить математические основы предмета и эффективные методы решения задач, входящих в состав экзаменационной работы за 11 класс (единый государственный экзамен)

Требования к результату освоения программы.

Предметные результаты:

1) сформированность представлений о роли информации и связанных с ней процессов в окружающем мире;

2) сформированность представлений о важнейших видах дискретных объектов и об их простейших свойствах, алгоритмах анализа этих объектов, о *кодировании и декодировании данных* и причинах искажения данных при передаче;

3) систематизация знаний, относящихся к *математическим объектам информатики*; умение строить математические объекты информатики, в том числе логические формулы;

4) владение опытом построения и использования *компьютерно-математических моделей*, проведения экспериментов и статистической обработки данных с помощью компьютера, интерпретации результатов, получаемых в ходе моделирования реальных процессов; умение оценивать числовые параметры моделируемых объектов и процессов; сформированность представлений о необходимости *анализа соответствия модели* и моделируемого объекта (процесса);

5) сформированность представлений о способах хранения и простейшей обработке данных; умение пользоваться *базами данных* и справочными системами; владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними;

6) владение навыками *алгоритмического мышления* и понимание необходимости формального описания алгоритмов;

7) овладение понятием *сложности алгоритма*, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;

8) владение стандартными приемами *написания на алгоритмическом языке программы* для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ; использование готовых прикладных компьютерных программ по выбранной специализации;

9) владение *универсальным языком программирования высокого уровня* (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;

10) владение умением *понимать программы*, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц;

11) владение навыками и опытом *разработки программ* в выбранной среде программирования, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ.

Метапредметные результаты:

1) умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

2) умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;

3) владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;

4) готовность и способность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;

5) умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и

организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.

Личностные результаты:

1) готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;

2) эстетическое отношение к миру, включая эстетику научного и технического творчества;

3) осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов; отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем.

Материально-техническое обеспечение. Для реализации программы необходимо наличие:

- отдельного помещения (класса с посадочными местами и столами); занятия проводятся в учебном классе общей площадью не менее 25 м², с посадочными местами для группы обучающихся;
- компьютерной техники для работы со средами программирования и тестирующими системами. Для реализации программы необходимо обеспечить каждому обучающемуся индивидуальное рабочее место с доступом к сети интернет. Необходимое оборудование: персональные компьютеры (на базе процессора не ниже Pentium4), мультимедийный проектор, доступ к сети интернет.

Мониторинг результатов обучения и критерии оценки обучающихся. Мониторинг результативности обучения по программе включает в себя три группы показателей:

1) теоретическая подготовка и основные общеучебные компетенции (фиксация приобретенных ребенком в процессе освоения образовательной программы предметных и общеучебных знаний, умений, навыков);

2) практическая подготовка (освоение обучающимися принципов решения задач повышенной сложности, умение определять наиболее эффективные способы достижения результата; формирование умения понимать причины успеха / неуспеха учебной деятельности и способности конструктивно действовать даже в ситуациях неуспеха; овладение логическими действиями сравнения, анализа, синтеза, обобщения, классификации, установления аналогий и причинно-следственных связей);

3) участие обучающихся в олимпиадах и конкурсах по информатике и

программированию (региональные соревнования, дистанционные отборочные этапы соревнований, организованных высшими учебными заведениями).

Формы определения результативности обучения по программе:

– самостоятельное решение промежуточных контестов (задачи с автоматической проверкой решения тестирующими системами);

– письменная работа, составленная в соответствии с кодификатором государственного итогового экзамена по информатике.

Качественные критерии оценки результатов деятельности детского объединения:

– изменение уровня сформированности понятий курса – умений, навыков (с помощью педагога или самостоятельно);

– способность использовать полученные знания при решении межпредметных и прикладных задач (предметная область – математика, информатика, шифрование данных, оптимизация алгоритмов).

Количественные критерии:

– первичные баллы, которые обучающиеся набирают при решении заданий в соответствии с кодификатором единого государственного экзамена по предмету «Информатика»;

– сохранность контингента учащихся (статистический отчет);

– количество учащихся, принявших участие в школьном и муниципальном этапе олимпиады по информатике;

– количество учащихся, принявших участие в отборочных дистанционных этапах всероссийских соревнования по информатике и программированию.

Кадровый потенциал реализации программы. К преподаванию допускаются лица, имеющие высшее образование по профилю педагогической деятельности, а также стаж профессиональной деятельности не менее пяти лет.

УЧЕБНЫЙ ПЛАН

дополнительной общеобразовательной общеразвивающей программы

«Углубленное изучение информатики. Подготовка к сдаче единого государственного экзамена (11 класс)»

Первый год обучения

Инвариантный (общетеоретический) модуль

Модуль	Всего часов	В том числе	
		теория	практика
Алгоритмизация и основы программирования	**96**	**48**	**48**
Синтаксис Phyton. Основные алгоритмические конструкции (условие, цикл)	24	12	12
Одномерные и двумерные массивы. Алгоритмы обработки массивов	24	12	12
Рекурсия. Процедуры и функции. Динамическое программирование	24	12	12
Однопроходные алгоритмы, символьные строки и последовательностей. Решение задач повышенного уровня. Сложные типы данных	22	11	11
Итоговая аттестация	2	-	2

Второй год обучения

Инвариантный (общетеоретический) модуль

Модуль	Всего часов	В том числе	
		теория	практика
Информация. Кодирование информации	24	10	14
Системы счисления	24	10	14
Анализ информационных моделей. Адресация в сети	24	10	14
Математическая логика	46	23	23
Промежуточная аттестационная работа	2	-	2
Решение задач повышенного уровня сложности (Часть С)	24	12	12

Модуль	Всего часов	В том числе	
		теория	практика
Подготовка к государственной итоговой аттестации. Отработка практических навыков, разбор типовых ошибок	44	10	34
Итоговая аттестация	4	-	4
Итого:	**192**	**75**	**117**

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК*

Начало первого года обучения – 01.02.2018 г.
Конец первого года обучения– 30.05.2018 г.
Начало второго года обучения– 01.09.2018 г.
Конец второго года обучения – 31 мая 2019 г.

** Продолжительность непосредственно учебного блока – 52 недели.*

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН
Первый год обучения
Инвариантный (общетеоретический) модуль

№ п/п	Модуль / тема модуля	Всего часов	В том числе		Формы аттестации (контроля)
			теория	практика	
1	**Синтаксис Phyton. Основные алгоритмические конструкции (условие, цикл)**	**24**	**12**	**12**	**Тестирование** **Решение итогового контеста**
1.1	Синтаксис Phyton. Правила записи операторов	6	3	3	
1.2	Ветвление. Каскадные условия	6	3	3	
1.3	Циклы. Циклы со счетчиком	6	3	3	
1.4	Циклы с условием. Предусловие, постусловие	6	3	3	
2	**Одномерные и двумерные массивы. Алгоритмы обработки массивов**	**24**	**24**	**24**	**Тестирование** **Решение итогового контеста**
2.1	Одномерные массивы. Списки	6	3	3	
2.2	Алгоритмы обработки массивов. Алгоритмы поиска	6	3	3	
2.3	Алгоритмы обработки массивов. Алгоритмы сортировки	6	3	3	
2.4	Двумерные массивы. Матрицы	6	3	3	
3	**Рекурсия. Процедуры и функции. Динамическое программирование**	**24**	**12**	**12**	**Тестирование** **Решение итогового контеста**
3.1	Процедуры. Локальные и глобальные	6	3	3	

№ п/п	Модуль / тема модуля	Всего часов	В том числе		Формы аттестации (контроля)
			теория	практика	
	переменные				
3.2	Функции. Параметры функции	6	3	3	
3.3	Рекурсия. Принцип работы рекурсивных алгоритмов	6	3	3	
3.4	Динамическое программирование	6	3	3	
4	**Однопроходные алгоритмы, символьные строки и последовательностей. Решение задач повышенного уровня**	**22**	**12**	**12**	**Тестирование** **Решение итогового контеста**
4.1	Понятие линейной и квадратичной сложности алгоритма.	6	3	3	
4.2	Эффективность алгоритма. Эффективность по времен и по памяти.	6	3	3	
4.3	Принципы реализации однопроходных алгоритмов	6	3	3	
4.4	Последовательности и строки	4	2	2	
5	**Итоговая аттестация**	**2**	**-**	**2**	
Итого:		**96**	**48**	**48**	

СОДЕРЖАНИЕ МОДУЛЕЙ ПЕРВОГО ГОДА ОБУЧЕНИЯ

1.1 Синтаксис Phyton. Правила записи операторов (6 ч.).

Учащиеся должны знать / понимать:

– общие сведения о языке Python. Разница между интерпретируемыми и компилируемыми языками программирования;

– структура программы на языке Python;

– установка Python на компьютер. Режимы работы Python;

– структура программы на языке Python;

– правила записи и использования комментариев;

– правила записи операторов ввода/вывода, оператора присваивания;

– арифметические операторы. Приоритет выполнение арифметических операций;

– особенности использования операторов целочисленного деления и взятия остатка от деления.

Учащиеся должны уметь:

– выполнить установку и настройку среды программирования;

– выполнить простейшую программу в интерактивной среде;

– написать комментарии в программе;

– использовать автоматизированную тестирующую систему для проверки решения задач;

– писать программы для решения задач (линейные алгоритмы, использование операторов целочисленного деления).

1.2 Ветвление. Каскадные условия (6 ч.).

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Учащиеся должны знать / понимать:

– назначение условного оператора;

– возможные способы записи условного оператора.

Учащиеся должны уметь:

– использовать логические операторы при решении задач;

– использовать множественное ветвление и каскадные условия;

– записывать сложные условия, используя отдельные логические операции.

1.3 Циклы. Циклы со счетчиком (6 ч.).

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Примеры использования циклов. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные

циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Учащиеся должны знать / понимать:

– назначение и особенности использования цикла с параметром;
– формат записи цикла с параметром;
– примеры использования циклов различных типов.

Учащиеся должны уметь:

– определять вид цикла, наиболее удобный для решения поставленной задачи;
– использовать цикл со счетчиком;
– корректно задавать границы диапазона, в котором изменяется параметр цикла;
– определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

1.4 Циклы с условием. Предусловие, постусловие (6 ч.).

Оператор цикла с условием. Бесконечные циклы. Альтернативная ветка цикла while. Корректная инициализация и изменения значения счетчика. Принципы использования предусловия и постусловия.

Учащиеся должны знать / понимать:

– циклы с условием и их виды;
– правила записи циклов условием.

Учащиеся должны уметь:

– определять вид цикла, наиболее удобный для решения поставленной задачи;
– использовать цикл с условием;
– корректно формулировать условие завершение цикла;
– определять целесообразность применения и использовать цикла с условием для решения поставленной задачи.

2.1 Одномерные массивы. Списки (6 ч.).

Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range.

Учащиеся должны знать / понимать:

– способ описания списка;
– способ доступа к элементам списка;
– операции, выполняемые со списками.

Учащиеся должны уметь:

– описывать списки;
– вводить элементы списка;
– выводить элементы списка;

– выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;

– использовать вложенные списки.

2.2 Алгоритмы обработки массивов. Алгоритмы поиска (6 ч.).

Базовые алгоритмы: вычисление суммы элементов, подсчет количества элементов массива, удовлетворяющих заданному условию, нахождение максимального (минимального) элемента массива и его номера. Алгоритмы поиска: прямой поиск, двоичный (бинарный) поиск.

Учащиеся должны знать / понимать:

– принципы обработки данных, содержащихся в массиве;

– принципы работы алгоритма прямого поиска;

– принципы работы алгоритма бинарного поиска.

Учащиеся должны уметь:

– выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;

– решать задачи с ограничением количества переменных на алгоритмы обработки массивов.

2.3 Алгоритмы обработки массивов. Алгоритмы сортировки (6 ч.).

Алгоритмы сортировки массивов: метод вставки, метод выбора, метод «пузырька».

Учащиеся должны знать / понимать:

– принципы работы алгоритмов сортировки;

– принципы выбора оптимального метода сортировки массива в зависимости от типа поставленной задачи.

Учащиеся должны уметь:

– выполнять сортировку элементов в списке по возрастанию и убыванию

– выполнять сортировку элементов в списке по указанному в задаче критерию

– определять оптимальный метод сортировки массива.

2.4 Двумерные массивы. Матрицы (6 ч.).

Матрицы. Вложенные списки. Матрицы. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Учащиеся должны знать/понимать:

– способ описания кортежа;

– способ описания словаря;

– понятие множества;

– способы описания множества;

– операторы работы с множествами.

Учащиеся должны уметь:

- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

3.1 Процедуры. Локальные и глобальные переменные (6 ч.).

Область видимости переменных в пространстве имен. Правила видимости имен. Глобальные переменные. Локальные переменные. Особенности работы оператора присваивания.

Учащиеся должны знать / понимать:

- способы объявления локальных и глобальных переменных;
- порядок сопоставления имен переменных внутри программы.

Учащиеся должны уметь:

- определять случаи перекрытия видимости пространства имен в программе;
- корректно использовать операторы присваивания для локальных и глобальных переменных.

3.2 Функции. Параметры функции (6 ч.).

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Учащиеся должны знать / понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- принцип передачи параметров.

Учащиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

3.3 Рекурсия. Принцип работы рекурсивных алгоритмов (6 ч.).

Рекурсивные функции. Вычисление факториала. Вычисление последовательности Фибоначчи. Рекурсивный алгоритм перевода чисел в двоичную и восьмеричную систему счисления.

Учащиеся должны знать / понимать:

- понятие функции;
- способы описания функции;

– принципы структурного программирования;
– понятие локальных переменных подпрограмм;
– понятие формальных и фактических параметров подпрограмм;
– принцип передачи параметров.

Учащиеся должны уметь:

– создавать и использовать функции;
– использовать механизм параметров для передачи значений.

3.4 Динамическое программирование (6 ч.).

Динамические структуры данных. Списки. Стек, очередь, дек. Деревья. Графы. Динамическое программирование.

Учащиеся должны знать / понимать:

– понятие динамического массива;
– примеры динамических массивов;
– понятие реккурентного соотношения.

Учащиеся должны уметь:

– решать сложные задачи путем сведения к более простым подзадачам;
– формулировать реккурентные соотношения.

4.1 Понятие линейной и квадратичной сложности алгоритма. Сложные типы данных (6 ч.).

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение.

Учащиеся должны знать / понимать:

– определение и виды сложных типов данных;
– возможности сложных типов данных;
– основные функции работы со словарями, словарные методы;
– основные функции работы с множествами, правила применения операций над множествами.

Учащиеся должны уметь:

– создавать словари, использовать словарные методы и операции;
– создавать множества, выполнять допустимые операции над множествами;
– целесообразно выбирать и использовать сложные типы данных при решении задач;
– определять линейную и квадратичную сложность решения.

4.2 Эффективность алгоритма. Эффективность по времени и по памяти (6 ч.).

Определение сложности алгоритма. Понятие емкостной и временной сложности. Зависимость допустимой сложности решения от ограничений, указанных в условии задачи. Принципы оптимизации алгоритмов по времени. Принципы оптимизации алгоритмов по используемой памяти. Определение сложности рекурсивных алгоритмов. Емкостная сложность.

Учащиеся должны знать / понимать:

- понятие емкостной и временной сложности алгоритма;
- принципы расчета емкостной и временной сложности алгоритма;
- зависимость допустимой сложности решения от ограничений указанных в условии задачи.

Учащиеся должны уметь:

- оценивать алгоритмы с точки зрения объема расходуемой памяти;
- оценивать алгоритмы с точки зрения затрачиваемого времени
- оптимизировать решение до приемлемого уровня сложности с учетом ограничений условия задачи.

4.3 Принципы реализации однопроходных алгоритмов (6 ч.).

Оптимизация алгоритма по времени: переход от вложенных циклов однопроходной обработке последовательности. Использование методов динамического программирования при оптимизации решения. Использование рекурсии при оптимизации решения.

Учащиеся должны знать / понимать:

- определение однопроходного алгоритма;
- преимущества однопроходной обработки последовательности;
- математические методы, используемые для перехода от вложенных циклов к однопроходной обработке последовательности.

Учащиеся должны уметь:

- оптимизировать алгоритм решения задачи с учетом действующих по условию задачи ограничений.

4.4 Последовательности и строки (4 ч.).

Тип данных – строки. Особенности индексирования символов строки в Python. Операции над строками. Методы работы со строками: len, find, rfind, count, replace.

Учащиеся должны знать / понимать:

- особенности работы со строками в Python;
- методы преобразования строкового и числового типа данных;
- методы работы со строками;
- принципы прямого и обратного индексирования символов строки.

Учащиеся должны уметь:

- считывать, хранить и преобразовывать строки;

– использовать методы работы со строками при решении задачи;
– выполнять операции над множеством строк;
– выполнять операции сортировки и поиска в множестве строк.

5. Итоговая аттестация по результатам первого года обучения

1

На вход алгоритма подается натуральное число N. Алгоритм строит по нему новое число R следующим образом.
1. Строится двоичная запись числа N.
2. К этой записи дописываются справа еще два разряда по следующему правилу:
а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;
б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы цифр на 2.
Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R.
Укажите такое наименьшее число N, для которого результат работы алгоритма больше 125. В ответе это число запишите в десятичной системе счисления

2

Запишите число, которое будет напечатано в результате выполнения следующей программы.

Паскаль
```
var s, n: integer;
begin
  s := 0;
  n := 0;
  while s < 111 do
begin
    s := s + 8;
    n := n + 2
  end;
  writeln(n)
end.
```

Python
```
s = 0
n = 0
while s < 111:
  s = s + 8
  n = n + 2
print(n)
```

Си
```
#include <stdio.h>
int main()
{ int s = 0, n = 0;
  while (s < 111) {
    s = s + 8;
    n = n + 2;
  }
  printf("%d", n);
  return 0;
}
```

3

Ниже записаны две рекурсивные функции (процедуры): F и G.Сколько символов «звездочка» будет напечатано на экране при выполнении вызова F(11)?

Паскаль
```
procedure F(n:
integer);
begin
```

Python
```
def F(n):
  if n > 0: G(n - 1)
def G(n):
```

Си
```
void F(int n) {
  if (n > 0) G(n - 1);
}
```

<table>
<tr><td></td><td>
<table>
<tr>
<td><pre>
 if n > 0 then G(n -
1);
end;
procedure G(n:
integer);
begin
 writeln('*');
 if n > 1 then F(n -
3);
end;
</pre></td>
<td><pre>
print("*")
if n > 1: F(n - 3)
</pre></td>
<td><pre>
void G(int n) {
 printf("*");
 if (n > 1) F(n - 3);
}
</pre></td>
</tr>
</table>
</td></tr>
<tr><td>4</td><td>Исполнитель Редактор получает на вход строку цифр и преобразовывает ее. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.
1. заменить (v, w)
2. нашлось (v)
Первая команда заменяет в строке первое слева вхождение цепочки v на цепочку w, вторая проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь».
Какая строка получится в результате применения приведенной ниже программы к строке, состоящей из 68 идущих подряд цифр 8? В ответе запишите полученную строку.
НАЧАЛО
ПОКА нашлось (222) ИЛИ нашлось (888)
ЕСЛИ нашлось (222)
ТО заменить (222, 8)
ИНАЧЕ заменить (888, 2)
КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ</td></tr>
<tr><td>5</td><td>В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Значения элементов равны 4, 7, 3, 8, 5, 0, 1, 2, 9, 6 соответственно, т.е. A[0] = 4, A[1] = 7 и т.д. Определите значение переменной c после выполнения следующего фрагмента этой программы.
<table>
<tr><th>Паскаль</th><th>Python</th><th>Си</th></tr>
<tr>
<td><pre>
c := 0;
for i := 1 to 9 do
 if A[i] < A[0] then
begin
  c := c + 1;
  t := A[i];
  A[i] := A[0];
</pre></td>
<td><pre>
c = 0
for i in
range(1,10):
 if A[i] < A[0]:
  c = c + 1
  t = A[i]
  A[i] = A[0]
</pre></td>
<td><pre>
c = 0;
for (i = 1;i < 10;i++)
 if (A[i] < A[0]) {
  c++;
  t = A[i];
  A[i] = A[0];
  A[0] = t;
</pre></td>
</tr>
</table>
</td></tr>
</table>

	A[0] := t; end;	A[0] = t	}

6 Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *M*. Известно, что *x* > 100. Укажите наименьшее такое (т.е. большее 100) число *x*, при вводе которого алгоритм печатает 26.

Паскаль	Python	Си

var x, L, M: integer;
begin
readln(x);
L := x;
M := 65;
if L mod 2 = 0 then
M := 52;
while L <> M do
if L > M then
L := L - M
else
M := M – L;
writeln(M);
end.
	x = int(input())
	L = x
	M = 65
	if L % 2 == 0:
	M = 52
	while L != M:
	if L > M:
	L = L - M
	else:
	M = M - L
	print(M)
		#include <stdio.h>
		void main()
		{
		int x, L, M;
		scanf("%d", &x);
		L = x; M = 65;
		if (L % 2 == 0) M = 52;
		while (L != M) {
		if(L > M)
		L = L - M;
		else
		M = M - L;
		}
		printf("%d", M);
		}

7 Напишите в ответе наименьшее значение входной переменной k, при котором программа выдает тот же ответ, что и при входном значении k = 10.

Паскаль	Python	Си

var k, i : longint;
function f(n:
longint): longint;
begin
f := n * n * n;
end;
function g(n:
longint): longint;
begin
g := 2*n + 3;
end;
begin
readln(k);
i := 1;
while f(i) < g(k) do
i := i+1;
writeln(i)
end
	def f(n):
	return n*n*n
	def g(n):
	return 2*n + 3
	k = int(input())
	i = 1
	while f(i) < g(k):
	i+=1
	print (i)
		#include <stdio.h>
		long f(long n) {
		return n * n * n;
		}
		long g(long n) {
		return 2*n + 3;
		}
		int main()
		{
		long k, i;
		scanf("%ld", &k);
		i = 1;
		while(f(i) < g(k))
		i++;
		printf("%ld", i);
		return 0;
		}

8	Исполнитель Калькулятор преобразует число на экране. У исполнителя есть две команды, которым присвоены номера: 1. Прибавить 1 2. Умножить на 2 Программа для исполнителя Калькулятор – это последовательность команд. Сколько существует программ, для которых при исходном числе 2 результатом является число 29 и при этом траектория вычислений содержит число 14 и не содержит числа 25?
9	На обработку поступает положительное целое число, не превышающее 10^9. Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0. Программист написал программу неправильно.

Паскаль

```
var N, digit, sum:
longint;
begin
  readln(N);
  sum := 0;
  while N > 0 do
begin
    digit := N mod 10;
    if digit < 7 then
      sum := sum + 1;
    N := N div 10;
  end;
  writeln(digit)
end.
```

Python

```
N = int(input())
sum = 0
while N > 0:
  digit = N % 10
  if digit < 7:
    sum = sum + 1
  N = N // 10
print(digit)
```

Си

```
#include <stdio.h>
int main()
{
  int N, digit, sum;
  scanf("%d", &N);
  sum = 0;
  while (N > 0) {
    digit = N % 10;
    if (digit < 7)
      sum = sum + 1;
    N = N / 10;
  }
  printf("%d",digit);
  return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 456.
2. Приведите пример такого трехзначного числа, при вводе которого программа выдает верный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;
2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки

10

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от –10 000 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива. Например, для массива из пяти элементов: 6; 2; 9; –3; 6 – ответ: 4. Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Паскаль	Python	Си

const n = 20;
var a: array [1..n]
of integer;
i, j, k: integer;
begin
for i := 1 to n do
readln(a[i]);
...
end.
	# допускается также
	# использовать две
	# целочисленные
	# переменные j и k
	a = []
	n = 20
	for i in range(0, n):
	a.append(int(input()))
	...
		#include <stdio.h>
		#define n 20
		int main() {
		int a[n];
		int i, j, k;
		for (i=0; i<n; i++)
		scanf("%d",
		&a[i]);
		...
		return 0;
		}

11	В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передается положительное целое число – текущее показание прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь. Необходимо вычислить «бета-значение» серии показаний прибора – минимальное четное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удается, ответ считается равным –1. **Задача А.** Напишите программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Максимальная оценка за выполнение задания А – 2 балла. **Задача Б.** Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик). Входные данные представлены следующим образом. В первой строке задается число N – общее количество показаний прибора. Гарантируется, что N > 6. В каждой из следующих N строк задается одно положительное целое число – очередное показание прибора. **Пример входных данных**: 11, 12, 45, 5, 3, 17, 23, 21, 20, 19, 18, 17. Программа должна вывести одно число – описанное в условии произведение либо –1, если получить такое произведение не удается. **Пример выходных данных** для приведенного выше примера входных данных: 54

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН
Второй год обучения
Инвариантный (общетеоретический) модуль

№ п/п	Модуль / тема модуля	Всего часов	В том числе		Формы аттестации (контроля)
			теория	практика	
1.	**Информация. Кодирование информации**	24	10	14	Тестирование (задания ЕГЭ № 5,9,13)
2.	**Системы счисления**	24	10	14	Тестирование (задания ЕГЭ № 1,9,16)
3.	**Анализ информационных моделей. Адресация в сети**	24	10	14	Тестирование (задания ЕГЭ № 3,4,12,15)
3.1	Анализ информационных моделей	12	5	7	
3.2	Адресация в сети интернет	12	5	7	
4.	**Математическая логика**	46	23	23	Тестирование (задания ЕГЭ № 2,18,23)
5.	**Решение задач повышенного уровня сложности (Часть С)**	26	13	13	Тестирование (задания ЕГЭ – 25)
6.	**Подготовка к государственной итоговой аттестации. Отработка практических навыков, разбор типовых ошибок**	44	10	34	
7.	**Итоговая аттестация**	4		4	Демо-версия ЕГЭ

СОДЕРЖАНИЕ МОДУЛЕЙ ВТОРОГО ГОДА ОБУЧЕНИЯ

1. Информация. Кодирование информации (24 ч.).

Содержательный подход к измерению количества информации. Единицы измерения количества информации. Алфавитный подход к измерению количества информации. Кодирование текстовой информации. Кодирование и обработка графической информации. Кодирование звуковой информации. Расчет скорости передачи информации по каналу связи.

Учащиеся должны знать / понимать:

- единицы измерения количества информации;
- смысл содержательного подхода к измерению количества информации;
- смысл алфавитного подхода к измерению количества информации;
- принципы кодирования текстовой информации, различные виды кодировок;
- принципы кодирования графической информации;
- принципы кодирования звуковой информации.

Учащиеся должны уметь:

- рассчитывать информационный объем сообщения, используя принципы алфавитного подхода;
- рассчитывать информационный объем сообщения, используя принципы содержательного подхода;
- соотносить единицы измерения информации и приводить их к единому значению
- рассчитывать информационный вес изображения
- рассчитывать информационный вес звукового сообщения
- производить сравнение информационного объема сообщений.

2. Системы счисления (24 ч.).

Позиционные и непозиционные системы счисления. Двоичная система счисления. Арифметика двоичных чисел. Восьмиричная система счисления. Арифметика восьмиричных чисел. Шестнадцатиричная система счисления. Арифметика шестнадцатиричных чисел. Перевод чисел между двоичной, восьмиричной и шестнадцатиричной, используя десятичную в качестве промежуточной. Перевод чисел между двоичной, восьмиричной и шестнадцатиричной, используя двоичную в качестве промежуточной. Отрицательные числа в позиционных системах счисления. Дробные числа в позиционных системах счисления.

Учащиеся должны знать / понимать:

- принципы записи чисел в непозиционных и позиционных системах счисления;
- принципы записи чисел в позиционных и позиционных системах счисления;

– правила перевода чисел в различные системы счисления;

– правила сравнения чисел, приведенных в различных системах счисления.

Учащиеся должны уметь:

– выполнять операции перевода между различными системами счисления;

– выполнять операции сравнения над числами, приведенными в различных системах счисления;

– выполнять арифметические операции над числами в различных системах счисления.

3.1. Анализ информационных моделей. Адресация в сети (12 ч.).

Электронные таблицы. Основные типы и форматы данных. Относительные, абсолютные и смешанные ссылки. Графические информационные модели: графы, диаграммы, таблицы. Анализ информационных моделей. Ориентированные и неориентированные графы. Взвешенные графы. Поиск оптимального пути.

Учащиеся должны знать / понимать:

– понятие абсолютной относительной и смешанной ссылки в электронных таблицах;

– принцип изменения адресации при копировании формулы из ячейки таблицы;

– правила построения диаграммы на основе данных электронной таблицы;

– понятие неориентированного, ориентированного и взвешенного графа.

Учащиеся должны уметь:

– выполнять корректное копирование формул из ячеек электронной таблицы;

– рассчитывать итоговое значение ячейки электронной таблицы по приведенной формуле;

– корректно соотносить построенную диаграмму и исходные данные;

– рассчитывать количество возможных путей по графу, представленному с табличном или графическом виде;

– рассчитывать оптимальный путь по графу, представленному с табличном или графическом виде, в соответствии с условиями задачи.

3.2. Адресация в сети интернет (12 ч.).

Сетевые протоколы http и TCP/IP. Правила формирование адреса внутри сетевого протокола. Адрес узла. Адрес сети. Маска сети. Расчет количества возможных узлов в сети. Определение номера компьютера в сети.

Учащиеся должны знать / понимать:

- определение сетевого протокола;
- принципы работы сетевых протоколов http и TCP/IP;
- правила формирования адреса узла в сети;
- правила формирования адреса сети;
- принцип работы маски сети.

Учащиеся должны уметь:

- вычислять адрес сети;
- вычислять адрес узла в сети;
- вычислять неизвестное значение части адреса маски сети;
- определять количество узлов в сети.

4. Математическая логика (46 ч.).

Логические операции, таблицы истинности, приоритет логических операций. Логические величины в программировании. Законы алгебры логики. Тождественные преобразования логических выражений. Логические формулы и логические схемы. Логические элементы, их обозначение в компьютерной схемотехнике. Методы решения логических задач. Логические функции на области числовых значений.

Учащиеся должны знать / понимать:

- основные понятия алгебры высказываний (высказывание, действия над высказываниями, таблицы истинности);
- логические основы устройства компьютера (логические законы, правила преобразования логических функций и выражений, базовые логические элементы).
- понятие графа, основные элементы графа;
- способы решения содержательных логических задач;
- способы записи условия задачи.

Учащиеся должны уметь:

- строить таблицы истинности логических функций;
- минимизировать логические выражения на основе законов алгебры логики;
- строить простейшие логические схемы из базовых логических элементов;
- выбирать способ решения содержательной задачи;
- записывать условие задачи в соответствии с выбранным способом решения;

– решать задачу в соответствии с выбранным способом;
– применять основные логические законы для решения задачи алгебраическим способом;
– анализировать информацию, сравнивать и сопоставлять ее.

5. Решение задач повышенного уровня сложности (Часть С) (26 ч.).

Элементы теории игр. Понятие выигрышной и проигрышной стратегии. Первая выигрышная позиция. Вторая выигрышная позиция. Дерево игры.

Учащиеся должны знать / понимать:
– определение выигрышной стратегии;
– определение проигрышной стратегии;
– принципы формирования выигрышной стратегии.

Учащиеся должны уметь:
– определять первую выигрышную позицию;
– определять вторую выигрышную позицию;
– строить дерево игры;
– обосновывать выбранную стратегию игры.

6. Подготовка к государственной итоговой аттестации. Отработка практических навыков, разбор типовых ошибок (44 ч.).

Критерии оценивания заданий 24-27. Типовые ошибки, допускаемые при решении заданий на понимание программного кода и проверку сложного условия (24). Типовые ошибки, допускаемые при решении задач на обработку массивов и последовательностей (25, 27). Типовые ошибки, допускаемые при решении задач на теорию игр массивов (26). Типовые ошибки, допускаемые при оптимизации алгоритма (27).

Учащиеся должны знать / понимать:
– критерии оценки развернутого ответа;
– правила оформления решения задания с развернутым ответом.

Учащиеся должны уметь:
– предварительно оценить количество баллов за предлагаемое решение с развернутым ответом.

7. Итоговая аттестация по результатам второго года обучения

1 (№ 32) Укажите наименьшее четырехзначное шестнадцатеричное число, двоичная запись которого содержит ровно 5 нулей. В ответе запишите только само шестнадцатеричное число, основание системы счисления указывать не нужно.

2 (№ 52) Логическая функция F задается выражением $(\neg x \wedge y \wedge z) \vee (\neg x \wedge \neg z)$. На рисунке приведен фрагмент таблицы истинности функции F, содержащий **все** наборы аргументов, при которых функция F истинна.

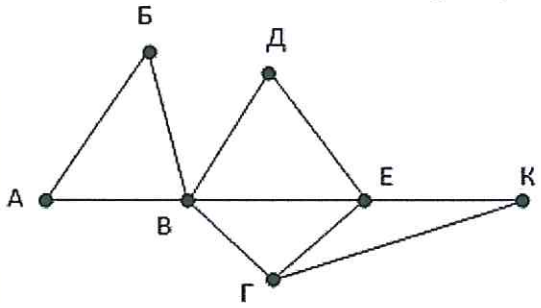
Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z.

Перем. 1	Перем. 2	Перем. 3	Функция
???	???	???	F
0	0	0	1
1	0	0	1
1	1	0	1

В ответе напишите буквы x, y, z в том порядке, в котором идут соответствующие им столбцы (без разделителей).

3 (№ 74) На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

	П1	П2	П3	П4	П5	П6	П7
П1		45		10			
П2	45			40		55	
П3					15	60	
П4	10	40				20	35
П5			15			55	
П6		55	60	20	55		45
П7				35		45	



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населенных пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта Г в пункт Е.

4 (№ 5) В каталоге находится 6 файлов:

maveric.map
maveric.mp3
taverna.mp4
revolver.mp4
vera.mp3
zveri.mp3

Ниже представлено восемь масок. Сколько из них таких, которым соответствуют ровно четыре файла из данного каталога?

ver.mp* *?ver?*.mp? ?*ver*.mp?* *v*r*?.m?p*
???*???.mp* ???*???.m* *a*.*a* *a*.*p*

5 (№ 111) Для передачи данных используется 5-битный код. Сообщение содержит только буквы А, Б и В, которые кодируются следующими кодовыми словами:

А – 11111, Б – 00011, В – 00100

Любые два кодовых слова отличаются друг от друга не менее, чем в трех позициях. Поэтому если при передаче кода буквы произошла одна ошибка, можно считать, что передавалась буква, код которой отличается от принятого в одной позиции. Если принятое кодовое слово отличается от

кодовых слов букв А, Б и В более, чем в одной позиции, считается, что произошла ошибка, которую обозначают символом «*».
Декодируйте сообщение

00110 00000 11111 11010

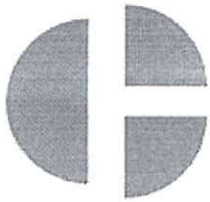
6 (№ 8) У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 2,
2. умножь на 5.

Запишите порядок команд в программе, которая преобразует **число 2 в число 24** и содержит не более четырех команд. Указывайте лишь номера команд.

7 (№ 10) Дан фрагмент электронной таблицы.

	A	B	C
1	???	6	10
2	=(A1-3)/(B1-1)	=(A1-3)/(C1-5)	=C1/(A1-3)



Какое целое число должно быть записано в ячейке A1, чтобы диаграмма, построенная по значениям ячеек диапазона A2:C2, соответствовала рисунку? Известно, что все значения ячеек из рассматриваемого диапазона неотрицательны.

8 (№ 168) Запишите число, которое будет напечатано в результате выполнения следующей программы.

Паскаль	Python	Си

var k, s: integer;
begin
k:= 5;
s:= 2;
while k < 120 do
begin
s:= s + k;
k:= k + 2;
end;
 	
write(s);
	k = 5
	s = 2
	while k < 120:
	s = s + k
	k = k + 2
	print(s)
		#include <stdio.h>
		int main()
		{ int k = 5, s = 2;
		while (k < 120) {
		s = s + k;
		k = k + 2;
		}
		
		printf("%d", s);
		return 0;
		}

end.		

9 (№ 13) Музыкальный фрагмент был записан в формате моно, оцифрован и сохранен в виде файла без использования сжатия данных. Размер полученного файла – 24 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате стерео (двухканальная запись) и оцифрован с разрешением в 4 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер файла в Мбайт, полученного при повторной записи.

10 (№ 205) Сколько слов длины 5, начинающихся с гласной буквы, можно составить из букв Е, Г, Э? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.

11 (№ 224) Алгоритм вычисления значения функции F(n), где n – натуральное число,
задан следующими соотношениями:

F(1) = 1
F(n) = F(n–1) * n, при n > 1
Чему равно значение функции F(5)?

12 (№ 244) По заданным IP-адресу узла сети и маске определите адрес сети:

IP-адрес: 10.8.248.131
Маска: 255.255.224.0

При записи ответа выберите из приведенных в таблице чисел 4 фрагмента четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без точек.

A	B	C	D	E	F	G	H
8	131	255	224	0	10	248	92

13 (№ 263) В школьной базе данных хранятся записи, содержащие информацию об учениках:

<Фамилия> – 16 символов: русские буквы (первая прописная, остальные строчные),

<Имя> – 12 символов: русские буквы (первая прописная, остальные строчные),

<Отчество> – 16 символов: русские буквы (первая прописная, остальные строчные),

<Год рождения> – числа от 1992 до 2003.

Каждое поле записывается с использованием минимально возможного количества бит. Определите минимальное количество байт, необходимое для кодирования одной записи, если буквы «е» и «ё» считаются совпадающими.

14 (№ 283) Исполнитель Редактор получает на вход строку цифр и преобразовывает ее.Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

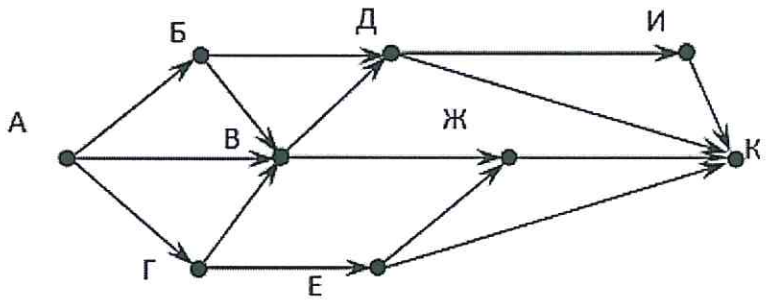
1. заменить (v, w)
2. нашлось (v)

Первая команда заменяет в строке первое слева вхождение цепочки v на цепочку w, вторая проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь».

Какая строка получится в результате применения приведенной ниже программы к строке, состоящей из 247 идущих подряд цифр 5? В ответе запишите полученную строку.

```
НАЧАЛО
ПОКА нашлось (222) ИЛИ нашлось (555)
 ЕСЛИ нашлось (222)
  ТО заменить (222, 5)
  ИНАЧЕ заменить (555, 2)
 КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ
```

15 (№ 302) На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



16 (№ 321) Запись числа 30 в системе счисления с основанием N оканчивается на 0 и содержит 4 цифры. Чему равно основание этой системы счисления N?

17 (№ 340) В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	*Найдено страниц (тыс.)*
мезозой	50
кроманьонец	60
неандерталец	70
мезозой | кроманьонец	80
мезозой | неандерталец	100
неандерталец & (мезозой | кроманьонец)	20

Какое количество страниц (в тысячах) будет найдено по запросу *кроманьонец & (мезозой | неандерталец)?*

18 (№ 359) На числовой прямой даны три интервала: P=[10,15], Q=[5,20] и R=(15,25]. Определите наибольшую возможную длину отрезка A, при выборе которого выражения

$$(x \notin A) \rightarrow (x \in P) \text{ и } (x \in Q) \rightarrow (x \in R)$$

принимают различные значения при любых x.

19 (№ 387) В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Значения элементов равны 7; 3; 4; 8; 6; 9; 5; 2; 0; 1 соответственно, т.е. A[0]=7; A[1]=3 и т. д. Определите значение переменной j после выполнения следующего фрагмента программы, записанного ниже на разных языках программирования.

Паскаль	Python	Си

j := 0;
for k := 1 to 9
do
begin
if A[k] <=
A[1] then
begin
A[1] :=
A[k];
j := j + k
end
end;
	j = 0;
	for k in range(1,10):
	if A[k] <= A[1]:
	A[1] = A[k]
	j = j + k
		j = 0;
		for (k = 1; k<= 9;
		k++)
		{
		if (A[k] <= A[1])
		{
		A[1] = A[k];
		j = j + k;
		}
		}

20 (№ 406) Укажите наибольшее из таких чисел x, при вводе которых алгоритм печатает сначала 3, а потом 120.

Паскаль	Python	Си
var x, L, M: integer;	x = int(input())	#include <stdio.h> int main(void)

```
begin
  readln(x);
  L:=0; M:=1;
  while x > 0 do
begin
    L:=L+1;
    M:= M*(x mod
8);
    x:= x div 8;
  end;
  writeln(L);
write(M);
end.
```

```
L = 0
M = 1
while x > 0 :
  L = L+1
  M = M*(x
% 8)
  x = x // 8
print(L)
print(M)
```

```
{
 int L, M, x;
 scanf("%d", &x);
 L = 0; M = 1;
 while (x > 0) {
  L = L + 1;
  M = M*(x % 8);
  x = x / 8;
 }
 printf("%d\n%d", L,
M);
}
```

21 (№ 425) Напишите в ответе наименьшее значение входной переменной k, при котором программа выдает ответ 21.

Паскаль	Python	Си

var k, i : longint;
function f(n:
longint): longint;
begin
  f := n * n * n;
end;
function g(n:
longint): longint;
begin
  g := n * n;
end;
begin
  readln(k);
  i := 1;
  while f(i) <=
k*g(i) do
  i := i+1;
  writeln(i)
end.
	def f(n):
	  return n * n
	* n
	def g(n):
	  return n * n
	k =
	int(input())
	i = 1
	while f(i) <=
	k*g(i):
	  i+=1
	print (i)
		#include <stdio.h>
		long f(long n) {
		  return n * n * n;
		}
		long g(long n) {
		  return n * n;
		}
		int main()
		{
		long k, i;
		scanf("%ld", &k);
		i = 1;
		while(f(i) <= k*g(i))
		  i++;
		printf("%ld", i);
		return 0;
		}

22 (№ 445) Исполнитель Калькулятор преобразует число на экране. У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Умножить на 2

Программа для исполнителя Калькулятор – это последовательность команд. Сколько существует программ, для которых при исходном числе 1

результатом является число 21 и при этом траектория вычислений содержит число 10?

23 (№ 469) Сколько существует различных наборов значений логических переменных x_1, x_2, ... x_{10}, которые удовлетворяют всем перечисленным ниже условиям?

$$(\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) = 0$$
$$(\neg x_2 \wedge x_3 \wedge \neg x_4) \vee (\neg x_2 \wedge x_3 \wedge x_4) \vee (x_2 \wedge \neg x_3 \wedge \neg x_4) = 0$$
$$\ldots$$
$$(\neg x_8 \wedge x_9 \wedge \neg x_{10}) \vee (\neg x_8 \wedge x_9 \wedge x_{10}) \vee (x_8 \wedge \neg x_9 \wedge \neg x_{10}) = 0$$

24 (№ 488) На обработку поступает положительное целое число, не превышающее 10^9. Нужно написать программу, которая выводит на экран количество цифр в десятичной записи этого числа. Программист написал программу неправильно.

Паскаль	Python	Си

var N: longint;
cnt: integer;
begin
readln(N);
cnt := 0;
while N > 1 do begin
cnt:=cnt + N mod 10;
N := N div 10;
end;
 	
writeln(cnt);
end.
	N = int(input())
	cnt = 0
	while N > 1:
	cnt = cnt + N % 10
	N = N // 10
	print(cnt)
		#include <stdio.h>
		int main()
		{
		int N, cnt;
		scanf("%d", &N);
		cnt = 0;
		while (N > 1) {
		cnt = cnt + N % 10;
		N = N / 10;
		}
		
		printf("%d",cnt);
		return 0;
		}

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 148.
2. Приведите пример такого трехзначного числа, при вводе которого программа выдает верный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:
   a. выпишите строку, в которой сделана ошибка;

b. укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

25 (№ 507) Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести максимальное значение среди двузначных элементов массива, не делящихся на 3. Если в исходном массиве нет элемента, значение которого является двузначным числом и при этом не кратно трем, то выведите сообщение «Не найдено».

Паскаль	Python	Си

const n = 40;
var
a: array [1..n]
of integer;
i, j, max: integer;
begin
for i := 1 to n do
 	
readln(a[i]);
...
end.
	# допускается также
	# использовать две
	# целочисленные
	# переменные j и max
	a = []
	n = 40
	for i in range(n):
	
	a.append(int(input()))
	...
		#include <stdio.h>
		#define n 40
		int main() {
		int a[n];
		int i, j, max;
		for (i = 0; i < n; i++)
		
		scanf("%d", &a[i]);
		...
		return 0;
		}

26 (№ 526) Два игрока, Паша и Вася, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу **один или три камня** или увеличить количество камней в куче **в два раза**. Игра завершается в тот момент, когда количество камней в куче становится не менее **33**. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 33 или больше камней. В начальный момент в куче было S камней, $1 \le S \le 32$.

Задание 1. а) Укажите все такие значения числа S, при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S, и укажите выигрывающий ход для каждого указанного значения S. б)Укажите такое значение S, при котором Паша не может выиграть за один ход, но при любом ходе Паши Вася может выиграть своим первым ходом. Опишите выигрышную стратегию Васи.

Задание 2. Укажите 3 таких значения S, при которых у Паши есть

выигрышная стратегия, причем Паша не может выиграть за один ход и может выиграть своим вторым ходом независимо от того, как будет ходить Вася. Для каждого указанного значения S опишите выигрышную стратегию Паши.

Задание 3. Укажите хотя бы одно значение S, при котором у Васи есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, и у Васи нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Васи. Постройте дерево всех партий, возможных при этой выигрышной стратегии Васи (в виде рисунка или таблицы).

Список литературы

Список основной литературы

1. Крылов С. С., Чуркина Т. Е. ЕГЭ-2018. Информатика и ИКТ. Типовые экзаменационные варианты. 10 вариантов. – М.: Национальное образование, 2017.
2. Крылов С. С., Чуркина Т. Е. ЕГЭ-2018. Информатика и ИКТ. Типовые экзаменационные варианты. 20 вариантов. – М.: Национальное образование, 2017.
3. Самылкина Н. Н., Синицкая И. В., Соболева В. В. ЕГЭ 2018. Информатика. Тематические тренировочные задания. – М.: Эксмо, 2017.
4. Самылкина Н. Н., Синицкая И. В., Соболева В. В., ЕГЭ 2018. Информатика. Сдаем без проблем! – М.: Эксмо, 2017.
5. Семакин И. Г. Информатика. Углубленный уровень: практикум для 10-11 классов: в 2 ч. Ч. 1 / И. Г. Семакин, Т. Ю. Шеина, Л. В. Шестакова. – М.: БИНОМ, Лаборатория знаний, 2013.
6. Угринович Н. Д Информатика и информационные технологии / Н. Д Угринович. – М.: Бином. Лаборатория знаний, 2017.
7. Угринович Н. Д. Информатика 10-11 класс / Н. Д. Угринович. – М.: Бином. Лаборатория знаний, 2017.
8. Угринович Н. Д. информатика и информационные технологии: Учебник для 10-11 классов / Н. Д. Угринович. – М.: Лаборатория Базовых Знаний, 2014.
9. Угринович Н. Д. Практикум по информатике и информационным технологиям / Н. Д. Угринович, Л. Л. Босова, Н. И. Михайлова. – М.: Бином. Лаборатория Базовых Знаний, 2013.

Список дополнительной литературы

1. Информатика: 9-11 классы: контрольные и самостоятельные работы по программированию / авт.-сост. А. А. Чернов, А. Ф. Чернов. – Волгоград: Учитель, 2009.
2. Семакин И. Г. Информатика и ИКТ. Профильный уровень: учебник для 10 класса / И. Г. Семакин, Т. Ю. Шеина, Л. В. Шестакова. – М.: БИНОМ, Лаборатория знаний, 2010.
3. Угринович, Н. Д. Информатика. Учебник для 7 класса / Н. Д. Угринович. – М.: БИНОМ. Лаборатория знаний; 2012.

Электронная поддержка образовательного процесса

1. Все о ЕГЭ: http://www.egeinfo.ru/.
2. Готов к ЕГЭ: http://www.gotovkege.ru/.
3. ЕГЭ по информатике: http://kpolyakov.spb.ru/.
4. ЕГЭ. Подготовка к ЕГЭ: http://www.ctege.org/.

5. Открытый банк заданий. Информатика: http://www.fipi.ru/.

6. Портал информационной поддержки единого государственного экзамена: http://ege.edu.ru/.

7. Российский образовательный портал Госэкзамен.ру: http://www.gosekzamen.ru/.

8. Российский общеобразовательный портал: http://www.school.edu.ru.